

Pause, Reflect and Act, the Pursuit of Continuous Transformation

Sandeep Hublikar^(✉) and Shrikanth Hampiholi^(✉)

CISCO Video Technologies India Pvt. Ltd., Bangalore, India
{sahublik, shampiho}@cisco.com

Abstract. Organizations take up agile transformation as silver bullet for all their business problems, but the fact is transformation journey is an eye opener to discover the real problems which were previously unnoticed. The authors were part of such a journey. It's easy to reap the obvious benefits of agile, but difficult to sustain and solve systemic obstacles like long build time, complex code base and legacy architecture that become a way of life over a long period of time. Here we describe the challenges we faced in sustaining our transformation beyond early victories and our efforts towards identifying and solving systemic obstacles across the organization by setting up an effective CI environment and addressing top people issues.

Keywords: Transformation · Agile · Sustenance · Scaling

1 Introduction

Video Business Unit (BU) in Cisco is a leader in Pay TV technology provider powering over 50+ Pay TV Service Providers and close to 80 million subscribers worldwide (and growing). It operates in cable, IP, mobile, terrestrial and satellite TV space. The BU has about 700 Engineers organized into more than 100 teams working on more than 40 projects based on a single code base. The authors of this paper are part of one project performing the role of Scrum master and Architect, additionally they are also part AGILE champions team responsible for deployment for AGILE practices across organization.

2 Background

CISCO Video Business unit, in order to position itself as the leading next generation broadcast platform, had to solve business challenges such as:

1. Disruptive technologies evolving in the Pay TV business
2. The need of employees to focus and have fun at work.
3. Improve predictability to launch a complex feature to our customers.

We had to change and change quickly to maintain and extend our competitive advantage.

Our legacy organization structure reflected our system architecture, where Teams were structured by components and subsystems, there were teams responsible for integration of these components and testing and validation responsibility was owned by a specialized team.

Analysis had repeatedly shown that multiple mutual dependencies and hand-offs between different teams slowed down deliveries, developers didn't feel responsibility for integration and testing which lead to late identification of defects and developers worked in silos which lead to local optimization instead of global optimization in terms of number of defects and performance.

After careful consideration and planning the agile transformation initiative was launched in year 2013. Once the decision was made the organization structure was changed to one suitable for Scrum [1]. Component teams were replaced by self-contained cross-functional feature development teams working from the common project code base (Grandmaster trunk). Developers, Testers, Integrators roles were all renamed to single role of developers. Similarly project leadership structure of Team Managers, Component Managers, Software Project Managers and Line Managers were replaced by Project Leadership Team (PLT) comprising of Product Owner (PO), Scrum Master (SM), Engineering Manager (EM) (Developers reported to EM) and Architect. Each of the 40+ projects was assigned their own Leadership team.

The first wave of agile transformation started to address problems of long requirement cycle and slow time to market. We focused on challenges of learning scrum practices, building CI machinery and cultivating a culture of delivering shipable code every sprint. In practice the customer deployed the software in field every quarter, the focus was to demonstrate and allow the customer to test and give early feedback on a sprint by sprint basis.

It was very challenging yet very interesting journey with a lot of opportunities on the way, a lot of learning and successful results. Slowly customers were acknowledging their happiness courtesy of the improved quality and on-time deliverables.

3 Ground Reality

Any transformational journey is a work in progress; either we keep improving or we start declining. Two years into the journey around middle of 2015 our transformation had hit a plateau and inefficiencies were creeping back in the organization.

Motivation levels were somewhat low thanks to some unresolved systemic obstacles like

- Long build times.
- Attrition of Subject Matter Experts with knowledge of Stack.
- Long learning curves for new comers due to complex codebase.
- Dependencies between different parties like driver providers, box manufacturers and chipsets vendors.
- Strict Definition of Done (DOD) without adequate supporting infrastructure.
- No safety net in terms automated test suite to avoid regression.
- Scalability and flexibility challenges in legacy architecture..

There were strong indications that at this rate, the transformation would fizzle out within a few quarters.

During a brain storming session involving the project leadership teams along with coaches and directors, metaphors were used to obtain a “pulse” or sentiment of the developers. One of the activities was to portray the biggest challenge being faced in our journey. Participants created an animal named ELIGA with small body and big/sharp teeth. (ELIGA is nothing but AGILE reversed). If not tamed at the earliest, this monster had the potential to eat and destroy whatever we had achieved so far (Fig. 1).



Fig. 1. ELIGA, an AGILE eating Monster

Since the start of journey we had learnt that multiyear timeframe is required for consistent sustainable agile transformation [2]. As transformation evolves business dynamics might change but organization would have embraced business agility.

Over a period, on time delivery of projects with agreed scope became priority over agile transformation. Secondly teams that were used to work in their own component specialization felt taxed in the new organization structure of self-sufficient cross functional teams.

Overall there was overwhelming consensus that over time the visible benefits of agile reaped by organization were going down.

4 Moving Forward

Multiple retrospections in the project leadership teams and feedback received from customers as well as developer community made it amply clear that periodic reinforcement of agile way of working for all stakeholders and executive commitment was a must for sustained transformation.

As a result we established an action team of about 10 people comprising Directors, Scrum Masters and coaches and put together a plan to re-energize and reestablish the organizational commitment to continuous transformation, thus BU wide *transformation 2.0* program was started. The authors of this report were part of the leadership team and active agile evangelists in the organization.

We evaluated Scaled Agile Framework (SAFe®) as most of projects were inter-linked and stacked on top of legacy projects. While SAFe® had its own merits we decided not pursue it as it didn't suit our context of running multiple independent projects on a common code base nor we were willing to invest in such large scale adoption again. We were practicing traditional scrum [1] in pockets and were quite happy with it.

We learnt that one of the better ways to solve the problems of the large program was by getting better at solving smaller, more focused problems. Interactions with industry practitioners and coaches had indicated that typically many large companies/accounts rush into trying to find big solutions to big problems, because they were not comfortable with improving the day-to-day activities, operational choices and obstacles which impact developers who are really doing the work. Becoming skillful at identifying small but obvious obstacles and resolving them had higher chance of resulting in the larger obstacles eventually fading away.

We wanted to find answers to the question: *“What does a happy organization look like and how do we get there?”*

In pursuit of finding answers to the above question a Vision Statement of the BU emerged which read as *“Happy People, Engineering Better Solutions, Everyday”*

We believed that a happy organization is the one where teams would deliver releases on time without stretching over weekends by:

- (a) Ensuring that an employee is motivated in their day to day work.
- (b) Manager is actively interested in employee's development.
- (c) Project Leadership team actively participates in Sprint Ceremonies.
- (d) Building skills for technical excellence (Engineering Excellence).
- (e) Directors actively attending sprint demos and appreciating team's Contribution.
- (f) Creating atmosphere of fun by celebrating small success

5 Action Plan

We had a series of workshops with the directors and leadership teams of all 40 + projects to identify the areas to focus in the near future and prioritize the epics to work on. The broad themes picked were *“Code Quality”* and *“Employee Engagement”*. These two epics were picked as priorities because these were major pain points and fixing them as fundamental for achieving further progress in transformation 2.0 (Fig. 2)

Some of the pain points with *“Code Quality”* were

- Frequently failing builds.
- Long cycle time to identify and fix regressions.
- Staying on Code repository branches for longer time due to release pressures.
- Long build time.

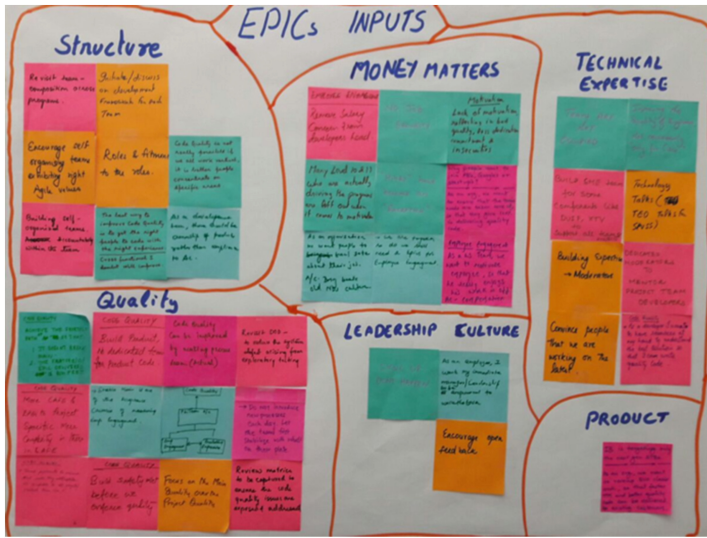


Fig. 2. Raw feedback as captured from the transformation 2.0 workshop

Some of the pain points with “Employee Engagement” were

- High attrition rate.
- Low scores and comments in engagement surveys.
- Lack of career path and role clarity.
- Lack of management involvement in project life cycle.

From these broad themes ten epics (Table 1) were derived and each epic had a sponsor who along with her core team ensured, supported and facilitated the implementation of the epic across organization. Each sponsor put together their own core teams (members of project leadership teams) who were passionate to work on the selected epics and to take the execution forward.

In the subsequent workshops, we arrived at plan to implement the epics across all customer projects, since each of the 40+ projects worked with their own backlog, each of the user stories from the epic became part of all the individual project backlogs. Respective product owners prioritized these user stories along with other user stories so that teams could plan in advance for the upcoming sprints without jeopardizing deliverables.

Epics ranging from infrastructure improvements like CI/CD to people centric improvements like Engagement and Leadership accountability were identified.

Table 1. Selected Themes split into Epics

Theme	Epic	Sponsor
Code Quality	Stable Code grandmaster main	Engineering Director 1
	Continuous Integration	
	Continuous Deployment	
	Building stack expertise	
Employee Engagement	Frequent interaction of Directors with scrum teams.	Engineering Director 2
	Participation in sprint ceremonies.	
	Proactive communication and timely resolution of obstacles.	
	Organize various forums to discuss issues common across organization	
	Monthly meeting with PO/SM/EM/Architects	
	Open house meetings in shorter groups	

Ways of working were agreed to drive these EPICs across the projects

1. Epic Sponsors would work with the core team to define and prioritize the user stories.
2. The sponsor and core team would work with the project leadership teams to drive the implementation in two week sprints.
3. Deploying the EPICS in projects was an added responsibility of the CORE team in addition to their regular project leadership work.
4. Release Demo to demonstrate what was the shift brought by these epics to be conducted at the end of 8 weeks.

6 Results

The Goal was to ensure everybody took steps together so that the positive change was felt uniformly across the whole system. This was one of core learning's from earlier transformation.

The table below summarizes the results we managed to achieve in the EPICs Chosen (Table 2).

Table 2. Status of epics so far achieved

	CODE QUALITY	EMPLOYEE ENGAGEMENT
Planned	<ul style="list-style-type: none">• Reduce Build Time.• Reduce Mean Time Between Failures (MTBF).• Improve success rate of builds.• Improve code coverage in sanity testing.	<ul style="list-style-type: none">• Informal interactions of Directors with the team.• Director's participation in Sprint demos.• Timely resolution of obstacles.• Periodic project retrospective across organization.

(Continued)

Table 2. *(Continued)*

	CODE QUALITY	EMPLOYEE ENGAGEMENT
Achieved	<ul style="list-style-type: none"> • Build time reduced from 48 Min to 23 Min. • MTBF reduced from many days to 24 h. • Build Success Rate increased to 80 %. • Sanity test Coverage increased to 96 %. 	<ul style="list-style-type: none"> • Regular formal one to one discussions between Directors and engineers. • Obstacle boards in Directors office. Directors pro-actively seeking acknowledgement of resolutions from submitters. • Reduced number of spill over user stories.
Challenges faced	<ul style="list-style-type: none"> • Ownership of build failures and identification of culprit check-in • Team capacity to take up transformation 2.0 user stories. 	<ul style="list-style-type: none"> • Project Leadership and accountability. • Detecting and Measuring RACI matrix [3]. • Shielding Developers from Customer escalations.
Future Goal	<ul style="list-style-type: none"> • Reducing MTBF to 2 h. • Developer level Ownership of build failures and fixing. • Improve Sanity to cover 100 %. • Focus on reducing regressions. • Double the number of builds per day on CI System. 	<ul style="list-style-type: none"> • Lower attrition rate. • Better scores and comments in surveys. • Happier faces around. • Developers approaching execs more often.

7 What We Learned

While it was very easy and straight forward to realize initial benefits of AGILE, beyond a point complacency sets in and its difficult to identify and improve unless there is a drive from top management. We also observed that the business continuity takes a front seat compared the commitment towards transformation which requires extreme courage and ability to take hard decisions that might be easy to take but are hard to live with. During the transformation journey it was very evident that it is far easier to implement and solve systemic obstacles than changing people mindset. Eg. It was easy to improve the CI/CD system as compared developing expertise or making people take ownership and being accountable.

Participation of people across the organization itself is a challenge when people are not clear of what the benefit to them is, even after lot of planning and persuasion, significant number of people in leadership team felt that there was no need for transformation 2.0 as we had improved on quality and timeliness of deliverables, also among the people who actively participated¹ there was a skew towards Scrum Masters and Engineering managers (Fig. 3).

¹ The figures show percentage of leadership team only as developers were not part of the transformation 2.0 driving committee.

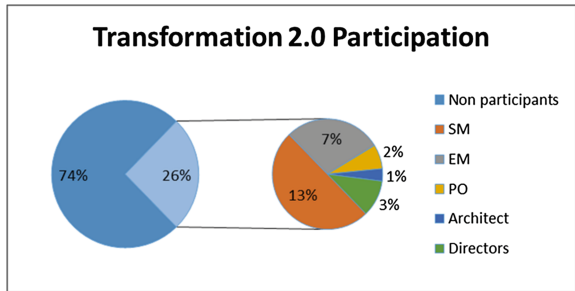


Fig. 3. Transformation Participation profile

Acknowledgements. We would like to thank CISCO for providing opportunity to contribute to agile transformation. We would like to thank colleagues for sharing agile learning's which have inspired this paper. We would also like to thank Ken Power for helping us to understand and appreciate rigor involved in submission of experience reports and thus helping us to prioritize. Last, but certainly not least, we would like to thank our experience report shepherd Maria Paasivaara for her gentle but firm guidance in shaping this experience report. *Thanks Maria for all the help. It probably couldn't have come together without you!*

Open Access. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, a link is provided to the Creative Commons license and any changes made are indicated.

The images or other third party material in this chapter are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt or reproduce the material.

References

1. CISCO Agile Play book
2. Large scale Agile Transformation at Danske bank, Innovate (2013)
3. Effective Project Management: Traditional, Agile, Extreme. Chapter 2, Robert K. Wysocki
4. Agile Product Development at Cisco. http://www.cisco.com/c/dam/en/us/products/collateral/customer-collaboration/unified-contact-center-enterprise/agile_product_development.pdf